Security for Real-Time MPEG Compressed Video in Distributed Multimedia Applications

George Anatassios Spanos† and Tracy Bradley Maples‡

† *The Aerospace Corporation*
‡ *California State University, Long Beach*

Abstract

Emerging computer network technologies promise to provide distributed multimedia in real-time. The security of real-time video, an important component of distributed multimedia applications, is increasingly becoming one of the major concerns of the computer networking community. This study presents Aegis, a security system for Motion Picture Experts Group (MPEG) compressed video transmissions.

Aegis utilizes the concepts of the MPEG video compression algorithm in order to provide an effective and efficient security mechanism. A prototype implementation is used to evaluate its effectiveness in securing video transmissions. Simulation results of Aegis in an Asynchronous Transfer Mode (ATM) technology environment are also presented in order to demonstrate its performance in comparison to traditional encryption/decryption methods.

1. Introduction

Undoubtedly, the saying "A picture is worth a thousand words" has characterized the evolution of communication media and technologies. Great inventions such as television, the VCR, and the video camera are clear examples of the human desire to interact with pictures, as well as with audio, and text. Computer hardware and software developers, realizing the effectiveness of multimedia in improving computer-to-human interaction, have strived to create a computer environment where video and sound can be combined with conventional data. The ability of computers to store, manipulate, and convey text, images, video, and audio has not only redefined the use of computers, but is now on the verge of revolutionizing human-to-human communications. The emergence of new computer network technologies that promise almost unlimited bandwidth, will allow millions of people to share in real-time video, audio, and textual information regardless of time constraints and geographical location.

Video conferencing, telemedicine, remote learning, concurrent engineering, and multimedia libraries are just a sampling of future multimedia applications. Corporations, which are expected to capitalize on the use of these new applications, are expecting to reduce travel, increase productivity, and compete more effectively in a global economy. Due to the critical nature of most of these applications, ensuring that multimedia transmissions are kept strictly private is vital to their use.

Data encryption/decryption has been the long-standing answer to security and authentication concerns. In recent years, several algorithms and techniques have been used to secure data and voice communications. Most of those techniques, however, were aimed at securing transmissions with low bandwidth, and delay requirements. Both encryption and decryption are computationally intensive operations, and although microprocessor speeds have logarithmically increased in the recent years, they remain time consuming processes.

TV-quality video, which is uncompressed and digitized, occupies about 8 Mbits per frame, and requires a bandwidth of 140-240 Mbps [1]. Advanced video compression techniques have dramatically reduced these requirements. For example, for MPEG-2 compressed video, a bandwidth of 5-30 Mbps and a delay of less than 150 ms can be achieved. Despite these improvements, it remains unknown if existing cryptographic algorithms are computationally fast enough for real-time video encryption.

In this paper, we take an alternative approach to video encryption by focusing on the characteristics of the data being transmitted. Video compression is becoming a vital factor in most video transmissions, and the *lossy* nature of most compression algorithms results in high sensitivity to errors and data loss. By selecting the most *sensitive* portions of a compressed video stream, we can reduce the amount of data to be encrypted, and

0-7803-3255-5/96 $5.00 © 1996 IEEE

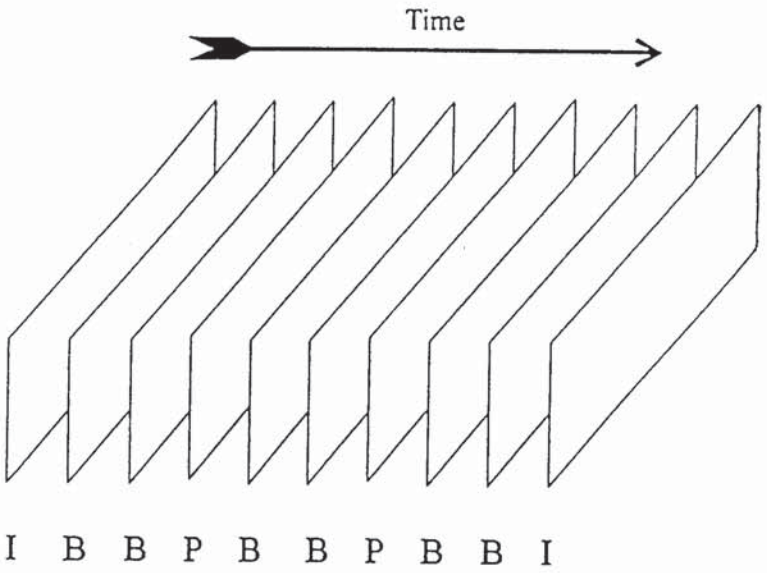

Fig. 1. MPEG "group of frames"

improve the enciphering/deciphering throughput. For the purposes of our study, we examine the Motion Picture Experts Group (MPEG) compression algorithm, because of its popularity and acceptance in the research community as the upcoming commercial video compression standard. We then explore Aegis, a security scheme which uses the Data Encryption Standard (DES) to selectively encrypt or decrypt all information necessary for the decompression of an MPEG compressed video stream.

2. MPEG Compressed Video

In the late-1980s, the International Organization for Standardization (ISO), in view of the increasing need for a standard for the storage of digital video and the associated audio, undertook an effort to develop a standard for storing such data on digital storage media, as well as, transmission on telecommunications channels such as ISDNs, and LANs. This ISO effort started in 1988 by the Moving Pictures Experts Group (MPEG), currently part of the ISO-IEC/JTC1/SC2/WG11.

MPEG is a *lossy* compression technique which retains only enough information for recovering the most significant parts of a compressed video stream [2]. In order to meet random access requirements without compromising quality requirements for high compression, MPEG introduced the concept of a "group of frames" (Figure 1). This group of frames contains a well balanced combination of both intraframe (stand-alone) and inter-frame coded frames (frames that contain picture changes in reference to a stand-alone frame). The intraframe coded "I" frames provide points for random access. High compression is maintained by use of predicted ("P") and interpolated ("B") interframe coded frames, which are based on adjacent "I" frames.

The MPEG video compression algorithm removes temporal redundancy using block-based motion compensation, and spatial redundancy using transform domain-(DCT) compression. Motion compensation, a technique also used in the CCITT standard H.261, is applied in two modes: prediction or interpolation. In the prediction mode, a current picture is modeled as a translation of a previous picture. Temporal redundancy is removed using motion compensated interpolation (also called bi-directional prediction), a unique feature of the MPEG algorithm. In interpolated pictures, a signal is obtained by adding a correction term to a combination of a past and a future reference. The resulting coded motion information is stored in 16x16 blocks, called macroblocks.

The images compressed with the application of motion compensation techniques, contain a very high spatial redundancy. MPEG chose to remove spatial redundancy using the Discrete Cosine Transformation (DCT) technique (this technique is also used in JPEG) with a combination of visually weighted scalar quantization and run length encoding. The 16x16 blocks obtained during the motion compensation

Table 1.--Syntax Layers of the MPEG video bit stream

Layer	Functionality
Sequence Layer	Random Access Unit: Context
Group of Pictures Layer	Random Access Unit: Video Coding
Picture Layer	Primary Coding Unit
Slice Layer	Resynchronization Unit
Macroblock Layer	Motion Compensation Unit
Block Layer	DCT Unit

process are used as inputs for the Discrete Cosine Transform which further compresses them, and produces a set of DCT coefficients. These coefficients are then quantized to achieve further compression by discarding information not visually significant. To further increase the remaining redundancy, the DCT coefficients are compressed using variable-length encoding. In this lossless compression phase, tables (similar to Huffman tables) are used to code events corresponding to a pair of {run, amplitude}. The events with high probability are coded using variable-length codes, while less-likely events are coded with an escape symbol followed by fixed length code in order to avoid long code words.

The resulting MPEG video bit stream contains six layers: each layer supporting a distinct function (Table 1). By defining the syntax of the compressed video stream, MPEG provides flexibility in the implementation of commercial encoders and decoders. In other words, MPEG considers a MPEG encoder to be any encoder that produces a video stream conforming to the specified MPEG syntax. Consequently, a MPEG decoder is any decoder that can decode a MPEG bit stream.

3. Aegis: Security for MPEG Video

3.1 Aegis Overview

Aegis exploits the great sensitivity of compressed video. The approach used by Aegis is the encryption of "I" frames for all MPEG groups of frames in a MPEG video stream. The choice of encrypting the "I" frames is based on the great significance of the intraframe (stand-alone) "I" frames in the decompression of a MPEG stream. "B" and "P" frames represent only translations of the picture information found in adjacent "I" frames; therefore, the encryption of "I" frames renders them useless (Figure 2). Furthermore, the intentional "corruption" of the stream has a serious impact on the outputs of the inverse DCT function during decoding. Recovery from such "corruption" is practically impossible.

In addition to the encryption of "I" frames, Aegis also encrypts the MPEG video sequence header. The sequence header contains all of the decoding initialization parameters such as the picture width, height, frame rate, bit rate and buffer size. The encryption of the sequence header, also conceals the MPEG identity of the stream and makes the MPEG video stream unrecognizable. In order to further conceal the MPEG identity of the stream, the ISO end code (last 32 bits of the MPEG video stream) is also encrypted.

Aegis' partial encryption of the MPEG video stream provides a high level of security because it presents two challenges to the ambitious network intruder. When portions of the MPEG stream are encrypted, the MPEG stream does not conform to the standard MPEG stream layered structure, and consequently, it is impossible to identify frames, groups of frames, or the encrypted "I" frames. The network intruder must first separate the encrypted portions of the stream, and then must still face the complexity of solving the encryption algorithm.

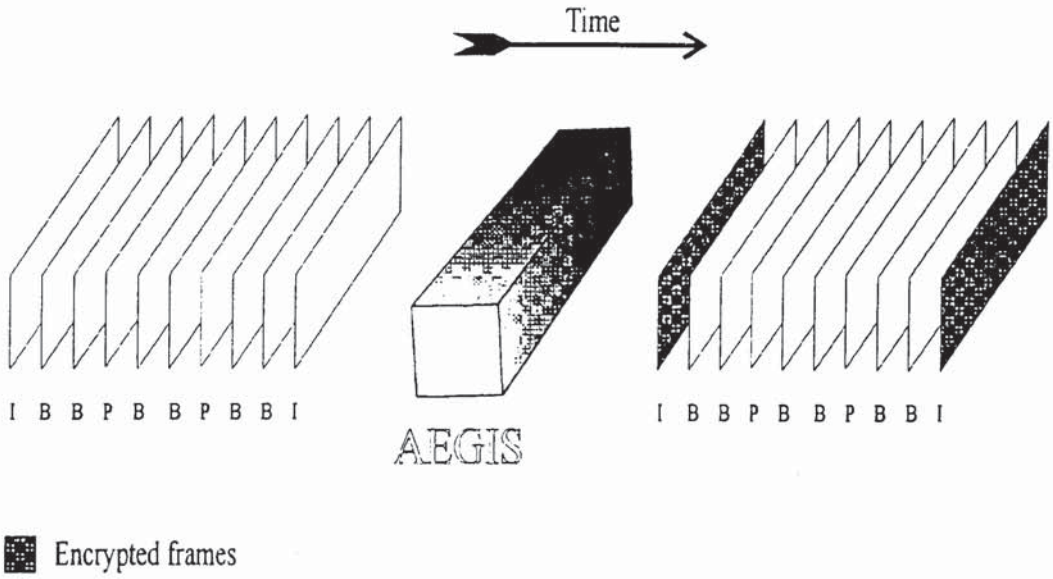

Fig. 2. Encryption of the stand-alone "I" frames

3.2 The Aegis Security Mechanism

Aegis acts as a stream filter between the MPEG encoder/decoder and the transmission mechanism. Although Aegis could be part of the MPEG encoder or decoder, as much independence as possible is maintained from the MPEG standard. Any direct relationship of the security mechanism and the encoding or decoding process would lead to the need for continuous adaptations of the security mechanism to any future MPEG modifications. Given the current scheme, the only dependence of the Aegis mechanism to the MPEG standard is knowledge of the MPEG stream syntax. The MPEG standard has maintained the concept of the "group of frames" through both the MPEG-1 and MPEG-2 proposals; therefore, we do not anticipate any conceptual changes in future MPEG compression enhancements that would force modifications to the Aegis security technique.

The MPEG system specification for the syntax of an MPEG stream, includes three coding layers (Figure 3): the ISO 11172 stream layer, the pack layer, and the packet layer. The ISO 11172 stream layer contains a sequence of packs followed by an end code. The pack layer includes a system synchronization clock reference (SCR), the multiplexing rate field, an optional header packet, and the packet layer. The packets that contain all information about the individual elementary streams, form the packet layer. Although the individual coding elements of the elementary streams are not byte-aligned, the packet contents and all system layer coding are byte-aligned. As a result the segments of "interest" can be easily distinguished by extracting groups of bytes, and comparing them against the start codes for sequence headers, groups of frames, and end codes.

One issue which arises when sections of the MPEG stream are encripted is how the encrypted section will be identified during the decryption process. Encryption algorithms alter slightly the size of the encrypted segments, and after the Aegis encryption process, the MPEG stream does not conform to the syntax discussed above. To facilitate the identification of the encrypted segments, Aegis employs flag-bytes using the bit-oriented technique of bit-stuffing. A flag byte, which is also kept secret to further complicate any cryptanalysis of the secured video stream, is inserted both at the beginning and at the end of each encrypted section. The bit stuffing technique ensures that the bit pattern chosen as the flag, does not appear between the starting and the ending flag by inserting a binary 0 digit following the first seven bits that correspond to the flag if they appear between the starting and ending flags. At the receiving end, after the starting flag is received, the process is reversed.

Aegis employs the DES encryption algorithm for the encryption process. Although a stream cipher may seem an attractive approach given the nature of the Aegis input, stream ciphers are considered weak and most have been successfully attacked. DES, on the

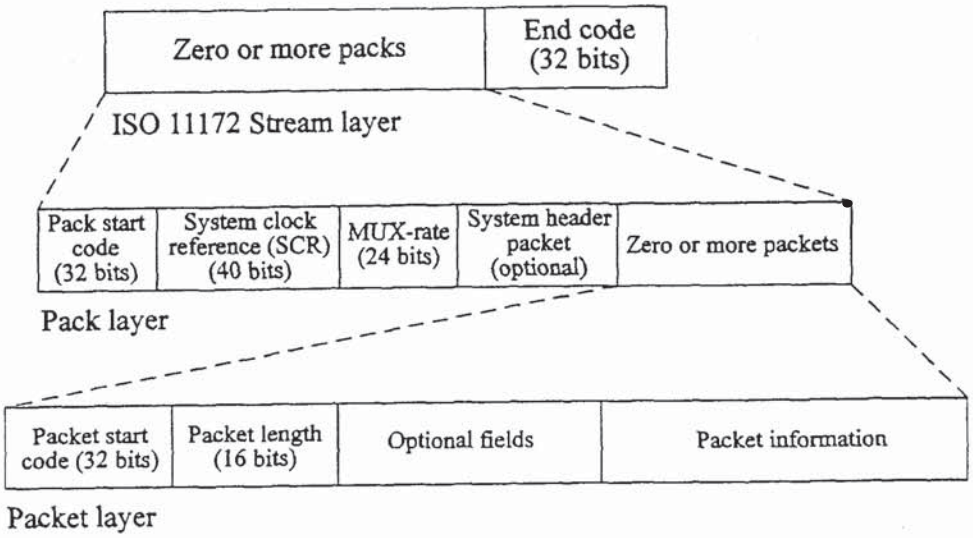

Fig. 3. The MPEG stream layers

other hand, is fast, flexible, easy to implement, and has been certified and validated. The block nature of the algorithm does not present any problems, since the data to be encrypted is extracted in groups of bytes. The extracted bytes can be arranged in groups of four, to accommodate the 64-bit block processing requirement of DES. DES is employed in the cipher block chaining mode (CBC). In this mode, a feedback mechanism feeds the results of the previous encryption to the current block. CBC is considered one of the most secure modes of DES, and it is the least affected by bit errors. Its ability to self-recover from bit-errors makes it very attractive for network applications, where bit errors are common due to noisy communication paths.

4. Aegis Performance and Conclusions

In order to obtain empirical results for the Aegis security scheme, a software implementation of Aegis was used. Aegis was coded using ANSI C on a 486-50 MHz personal computer. Segment and byte flag identification was performed at approximately 50 Mbps, while the encryption/decryption process exhibited a rate of approximately 2 Mbps.

A special MPEG video player was employed in order to empirically verify the security of MPEG video. Commercially available MPEG players exhibited fatal errors when they encountered the encrypted video stream segments. Our player, which employed an error-recovery mechanism that forced the display of frames that seemed *corrupted*, was successful in decoding and displaying the entire video stream. All video image objects in motion were scrambled beyond recognition. Background images, as it was expected from the availability of background information in the un-encrypted portions of the MPEG stream, appeared to slightly improve with time, but they also seemed blurry and barely recognizable. Samples of a MPEG video frame and the corresponding Aegis, and fully encrypted video frames are presented for comparison in Figure 4. From this sample, and many others like it, it seems probable that Aegis can be successfully used as designed: to provide an acceptable level of security for a MPEG video stream.

Since the Aegis mechanism was primarily conceived for use in integrated service networks, we also examined the benefits resulting from its use in an ATM technology network. Through a discrete-event computer simulation, we compared our proposed scheme to the encryption of all transmitted video frames. Assuming DES encryption and decryption at 10 Mbps, and using Studio TV traffic as input, Aegis was able to maintain a mean end-to-end delay on the order of a few milliseconds (Figure 5). In contrast, the full encryption scheme was unable to keep up with the input video stream. Its end-to-end delay never acheived a steady state, but rather increased throughout the simulation rum. These simulation results demonstrate that the Aegis mechanism can provide security for video streams while significantly out-performing conventional encryption and decryption processing in a distributed application environment. A detailed discussion of our simulation results can be found in [6].

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS
Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS
Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS
Sync your system to PACER to automate legal marketing.